How do I use the Host Interface?

The Host interface of the [TPS25750](https://www.ti.com/product/TPS25750) is implemented on top of the I2C bus protocol

If you are not familiar with the specific details of the I2C bus, I suggest reading the [I²C-bus Specification, Version 6.0, 4th of April 2014](https://www.nxp.com/docs/en/user-guide/UM10204.pdf)

The host interface on the [TPS25750](https://www.ti.com/product/TPS25750) as described in the [TPS25750 Host Interface Technical Reference Manual](https://www.ti.com/lit/pdf/slvuc05) consists of two possible operations

1. Reading and Writing Registers
2. Issuing 4CC commands
   1. The 4CC commands for the [TPS25750](https://www.ti.com/product/TPS25750) are based on an input data structure that includes a string processor. When creating the data structure to write the CMD1 register, the data passed into the CMD1 register to execute the command is the number of characters in the string and the actual string of ASCII characters that is described in the [TPS25750 Host Interface Technical Reference Manual](https://www.ti.com/lit/pdf/slvuc05)
   2. For example, the I2CR command you would write the following group of characters to the CMD1 register {4, ‘I’, ‘2’, ‘C’,’R’}

# Here is the I2C traffic view of a Register Write to the [TPS25750](https://www.ti.com/product/TPS25750).



I will walk through this graphic in detail so that you can fully understand to meaning of the blocks. I will not be explaining future blocks because they are based on this

The first portion of the diagram is the addressing phase for the I2Cs interface on the [TPS25750](https://www.ti.com/product/TPS25750) This value will be set via the ADCIN1 and ADCIN2 and can be any value between 0x20 and 0x23. Note that the I2C address is a 7 bit value that excludes the R/W bit from the I2C interface. The R/W bit in this case is a W, but for I2C read commands it will be called out as an R.

The next portion of the register access can be between 3 and 47 bytes. The first byte is the register address from the [TPS25750 Host Interface Technical Reference Manual](https://www.ti.com/lit/pdf/slvuc05), the second byte is the number of bytes between 1 and 64, and the next 1 to 64 bytes are the bytes to be written to the register

Here is an example of how to use the I2Cw command:



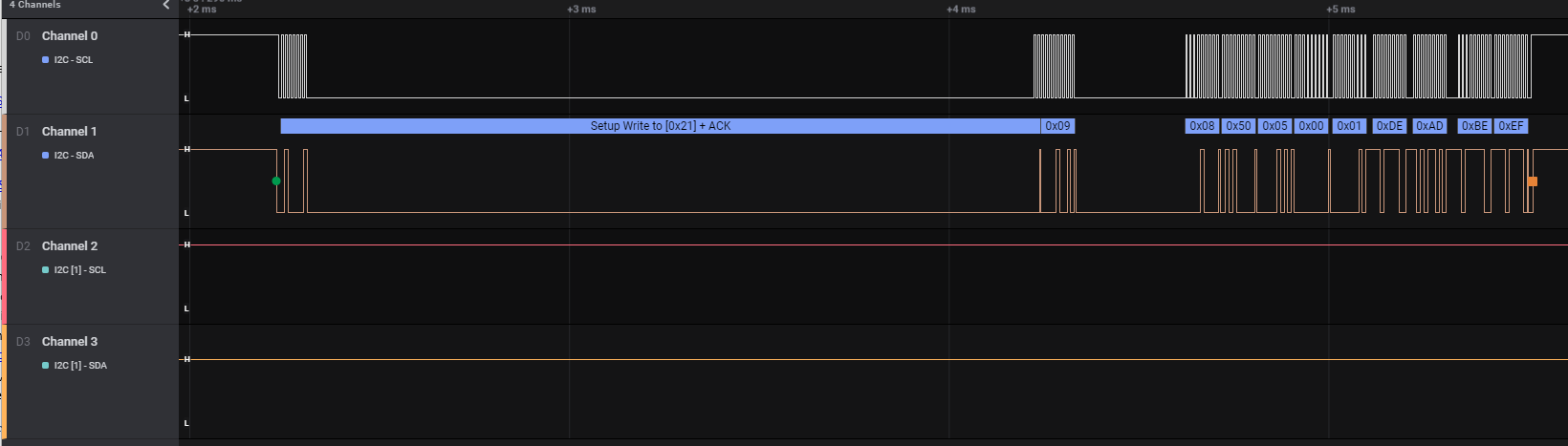


Figure 1: Data1 Register Update

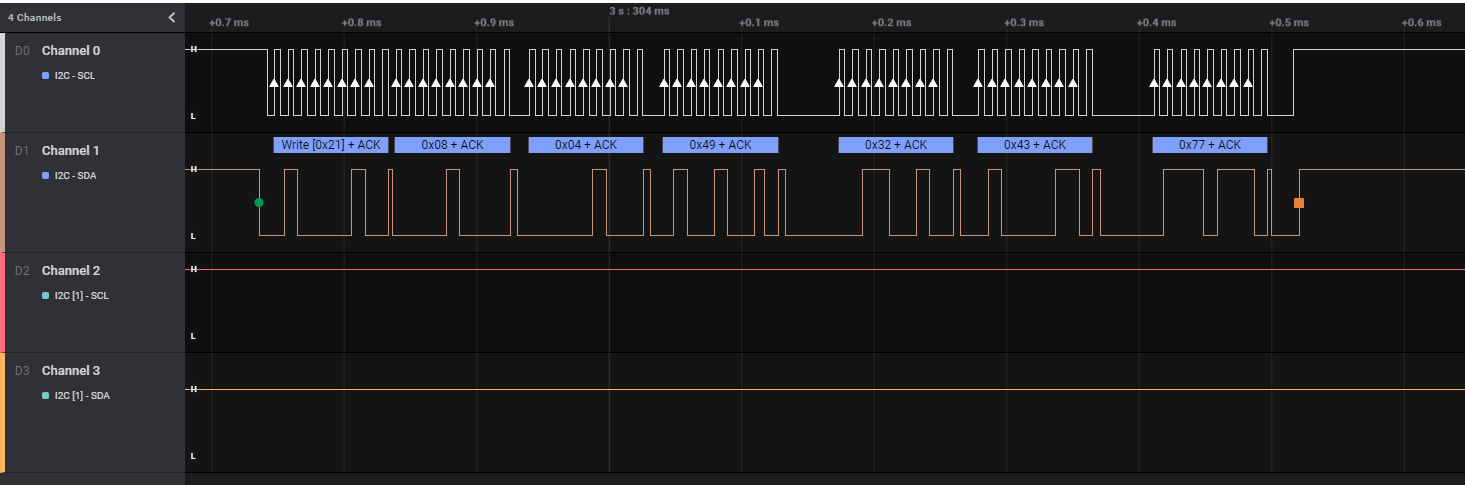


Figure 2 Write I2Cw to Command Register

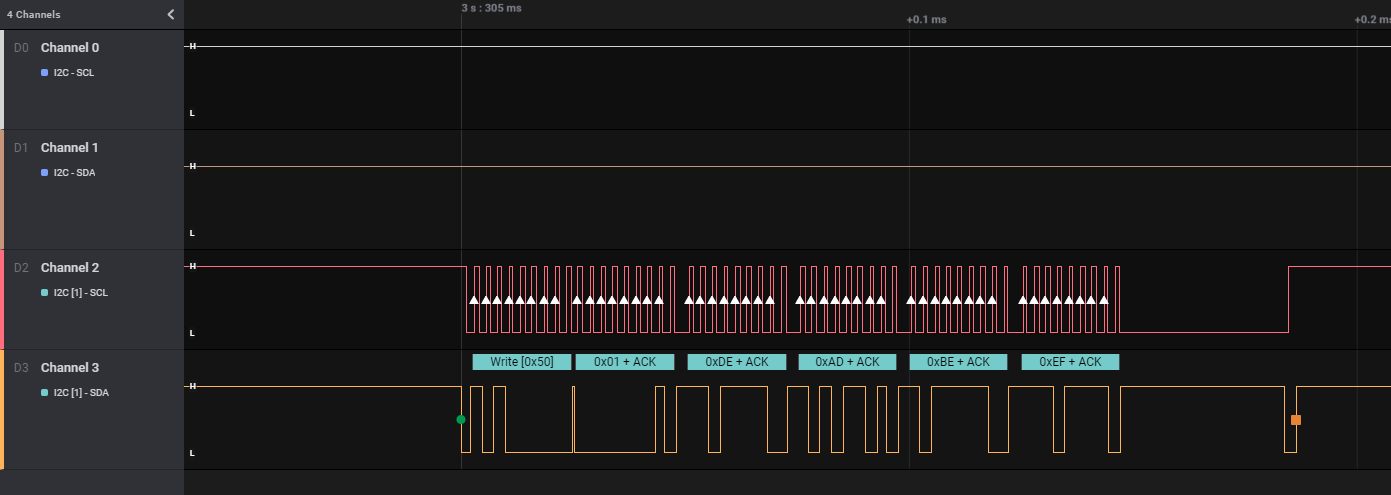


Figure 3 Resulting write to I2Cm Bus

If you were to create an API call to write this function, it could look something like

Int HI\_I2C\_WRITE(TPS255750Address, RegisterADDRESS, ByteCNT, ByteArray[63]);

The return code would be 0 for a successful write and not zero if the I2C bus NAK’d the write

# Here is the I2C traffic view of a Register Read from the [TPS25750](https://www.ti.com/product/TPS25750).



The register Read command for the [TPS25750](https://www.ti.com/product/TPS25750) is actually 2 I2C commands

The first command is an I2C write to set the Register Address to be Read



Followed by an I2C read command to read the data from the register



Not that this is the first time that the R bit is set in the chip addressing phase.

If you were to create an API call for this function, then it would look something like this

Byte\_array[65] HI\_I2C\_Read (TPS255750Address, RegisterADDRESS);

The return code would look something like this

Byte 0 would be the byte Count returned

Byte 1 would be the CMD response code

This will be 0x00 for a successful read and not zero for a failed read. For more information about failure codes, please refer to

Byte 2-65 will be the Bytes read from the [TPS25750](https://www.ti.com/product/TPS25750) the [TPS25750 Host Interface Technical Reference Manual](https://www.ti.com/lit/pdf/slvuc05)

Now Let’s build a Host Interface Commands using the Host Interface I2C register read and write Commands

The process to initiate the command requires writing 2 I2C registers.

The first register is the DATA1 register that is setup to provide all of the data necessary to execute the command. The second I2C register is the CMD1 register. This register is used by the string processor that will cause the [TPS25750](https://www.ti.com/product/TPS25750) to process the command.

The next step is to repeatedly read (polling) the CMD1 register to see that it changes from ‘I2CR’ to any other value. If the register return is 0x00, then the CMD was successfully executed. If it is any other value, then an error occurred

If the CMD1 return code is 0x00, then the next step is to read the DATA1 register and validate the data. The return data packet byte[1] will contain a return code. If this code is not 0x00, then an error occurred.

If the return code is 0x00, then the data is valid and can be used in your program

A graphical Example of this process is shown below

The other host interface commands described in the [TPS25750 Host Interface Technical Reference Manual](https://www.ti.com/lit/pdf/slvuc05) can be constructed using this same process with different data values and CMD strings.